ADAPTIVE TECHNIQUES FOR SCALABLE OPTIMISTIC PARALLEL DISCRETE EVENT SIMULATION

Eric Mikida

Presentation Overview

- PDES / GVT Overview
- GVT Framework Description
- New GVT Algorithm
- New Load Balancing Work
- Summary and Future Work

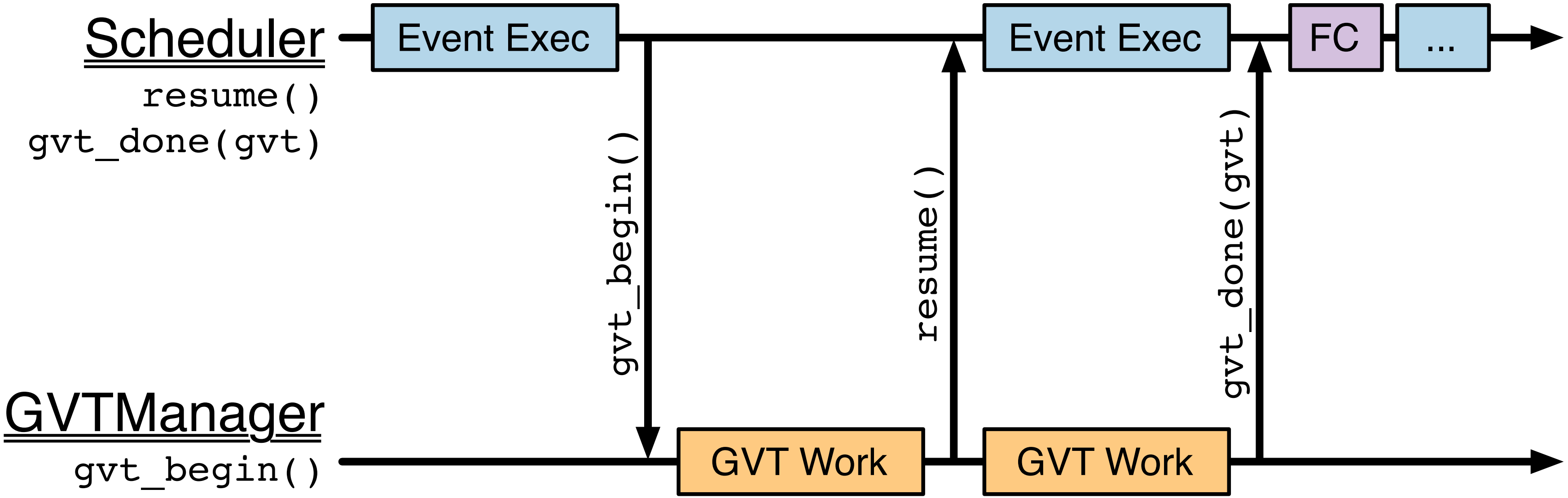
PDES / GVT Overview

- Simulation driven by discrete, time-stamped events
- Logical Processes (LPs) store state and execute events
- Charades is optimistically synchronized
  - Events executed speculatively
  - Incorrect events rolled back via reverse computation
  - Event efficiency = committed / total
- Global Virtual Time (GVT) required for synchronization
  - Virtual time passed by every processor and event in flight

GVT Framework Description

- Separated GVT Management from Scheduler
  - Each encapsulated into separate chare groups
  - Common API between base classes
- Allows for multiple different GVT implementations
- Work and communication automatically overlapped with Scheduler and LPs

GVT Framework Description



Scheduler

resume()

gvt_done(gvt)

GVTManager

gvt_begin()

Event Exec

gvt_begin()

GVT Work

resume()

Event Exec

GVT Work

gvt_done(gvt)

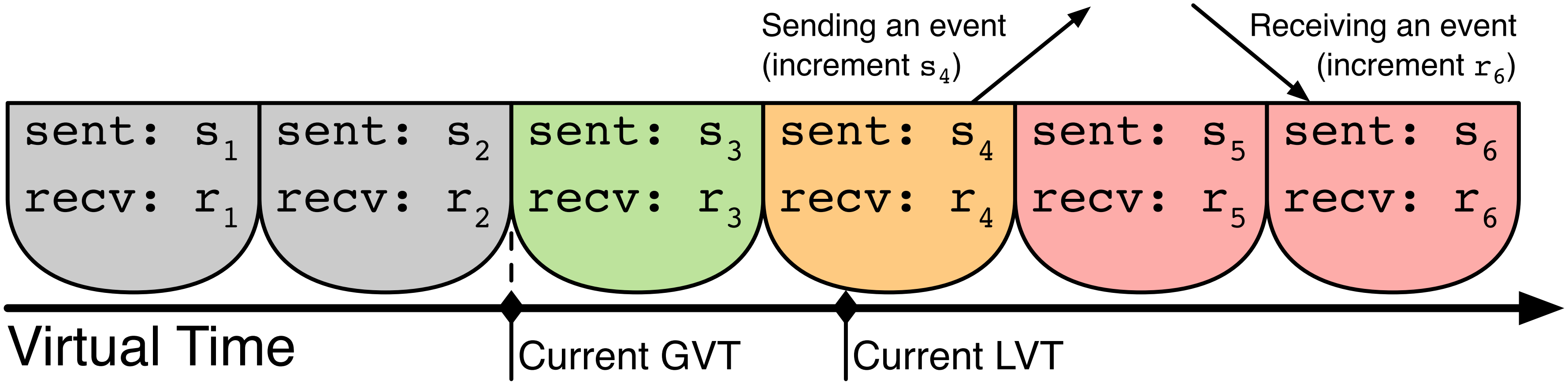
FC

...

New GVT Algorithm

- Adaptive Bucketed GVT algorithm
  - Virtual time divided into buckets
  - Completion detection per bucket
  - CD is timestamp aware
  - Buckets included in a given computation can increase/decrease based on simulation conditions

Adaptive Bucketed GVT Algorithm



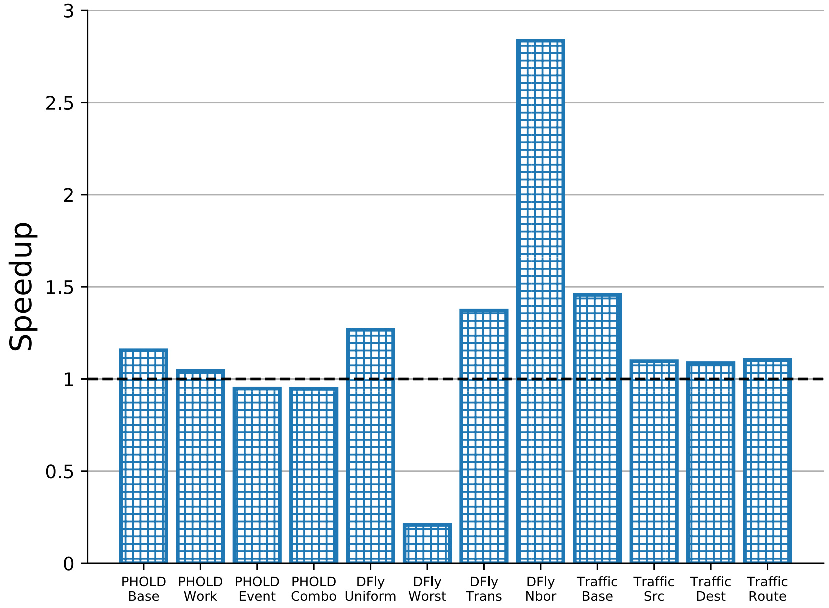
Adaptive Bucketed GVT Algorithm

Formally, a bucket *b* is completed iff:

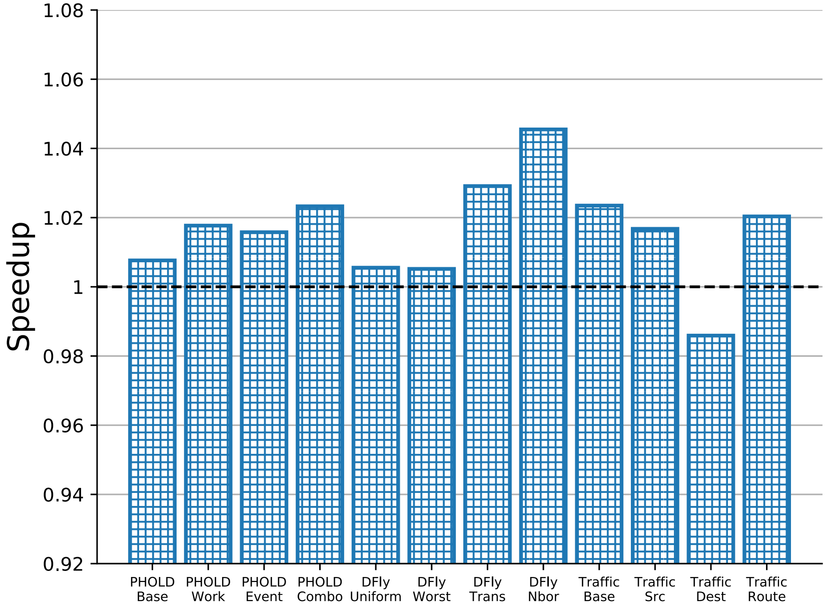
1) $\texttt{sent}[b] = \texttt{recvd}[b]$
2) $\texttt{lvt}_p > b \times \texttt{bucket\_size}$ for all processors *p*
3) bucket *x* is complete for all *x* in $\{ 1 \ldots b\text{-}1 \}$

Adaptive Bucketed Performance

Speedup over Blocking



Speedup over Phase-Based



Adaptive Bucketed Interval Analysis

All-Reduces for Phase-Based

	Total	Per GVT
PHOLD Base	3887	4.11
PHOLD Work	4270	4.31
PHOLD Event	5553	4.28
PHOLD Combo	6890	4.33
Traffic Base	--------	--------
Traffic Src	--------	--------
Traffic Dest	--------	--------
Traffic Route	--------	--------

All-Reduces for Adaptive Bucketed

	Total	Per GVT
PHOLD Base	2005	1.98
PHOLD Work	2024	1.98
PHOLD Event	2011	1.98
PHOLD Combo	2040	1.99
Traffic Base	1276	1.58
Traffic Src	1965	1.92
Traffic Dest	1350	1.57
Traffic Route	2027	1.99

Adaptive Event Throttling

- SPEEDES halted event sending to flush network for continuous GVT [1]
  - Execution was allowed to continue
- Anti events a source of significant overhead
- Adaptive Bucketed GVT monitors all event sends (both regular and anti events)

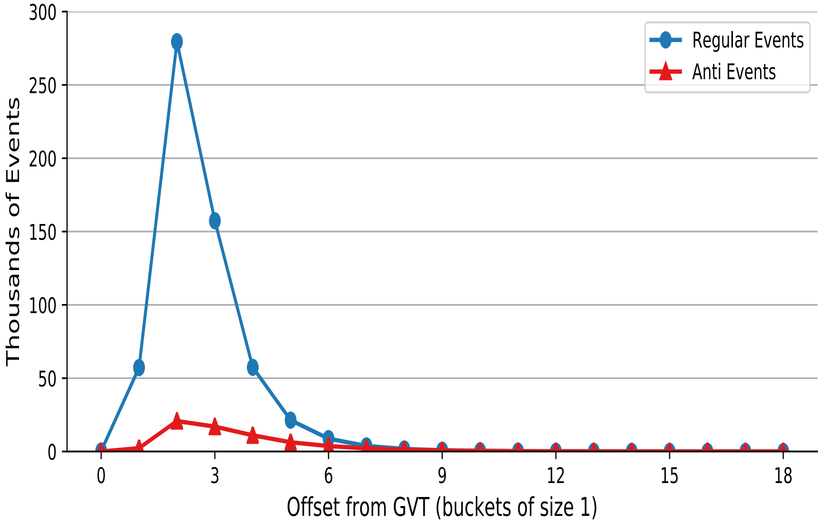
[1] Steinman ‘95

Adaptive Event Throttling

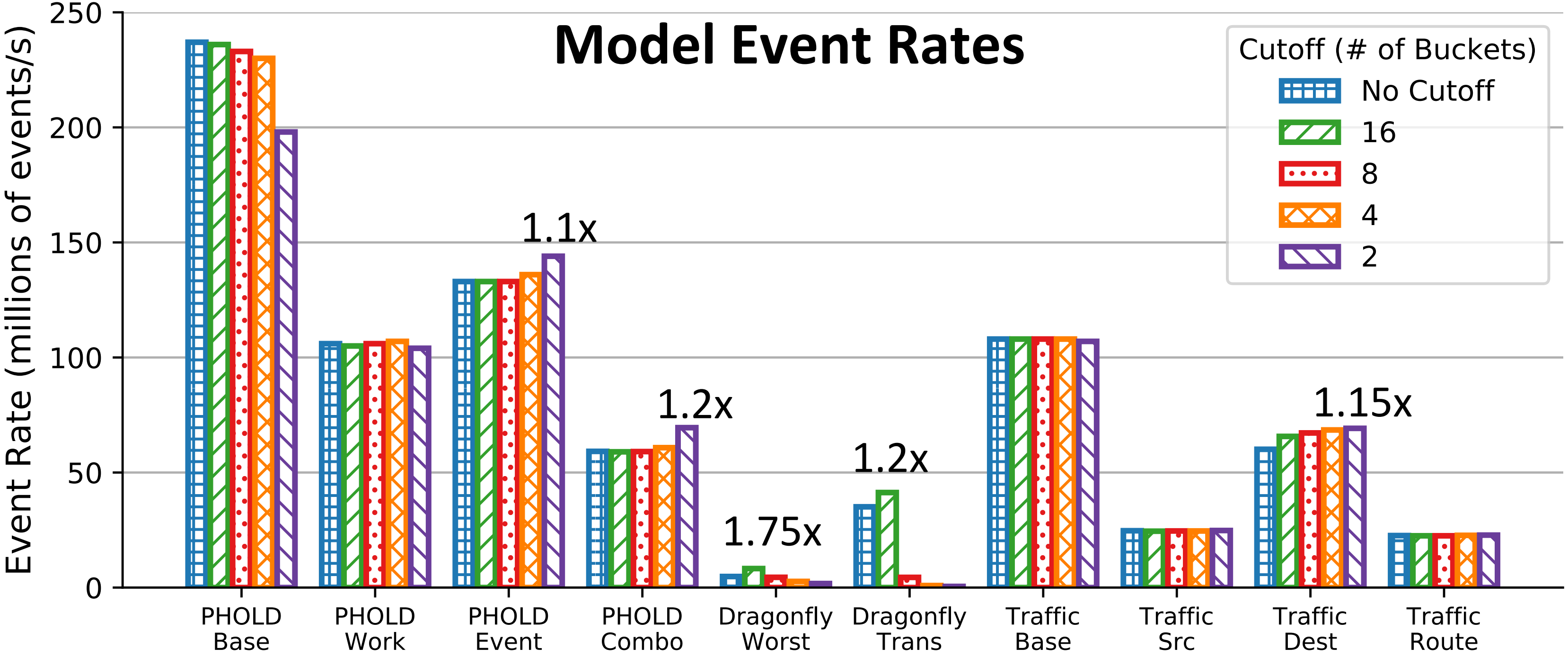
Approach

- Track events by offset from GVT (in buckets)
- Add tracing for off-line analysis
- Analyze cancellation frequency and lag
- Hold events based on offset

PHOLD Combo Event Stats

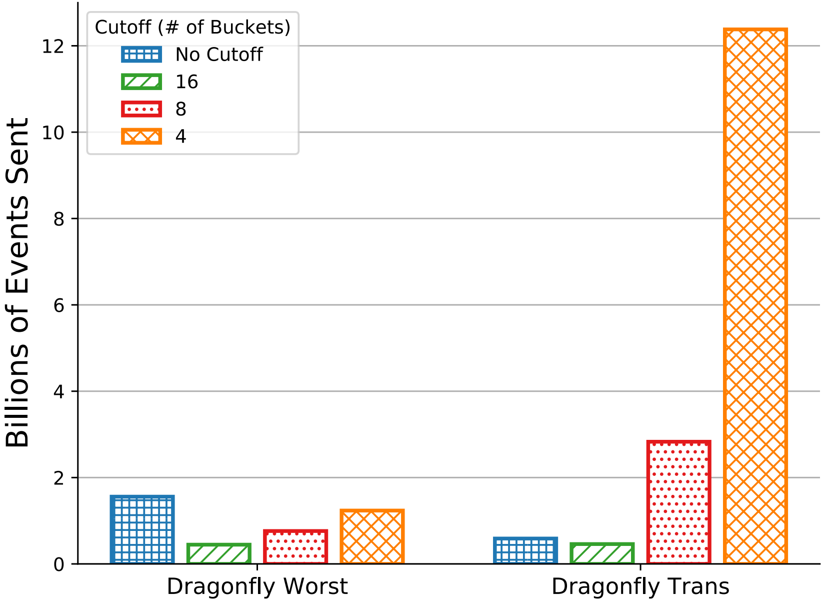


Adaptive Event Throttling

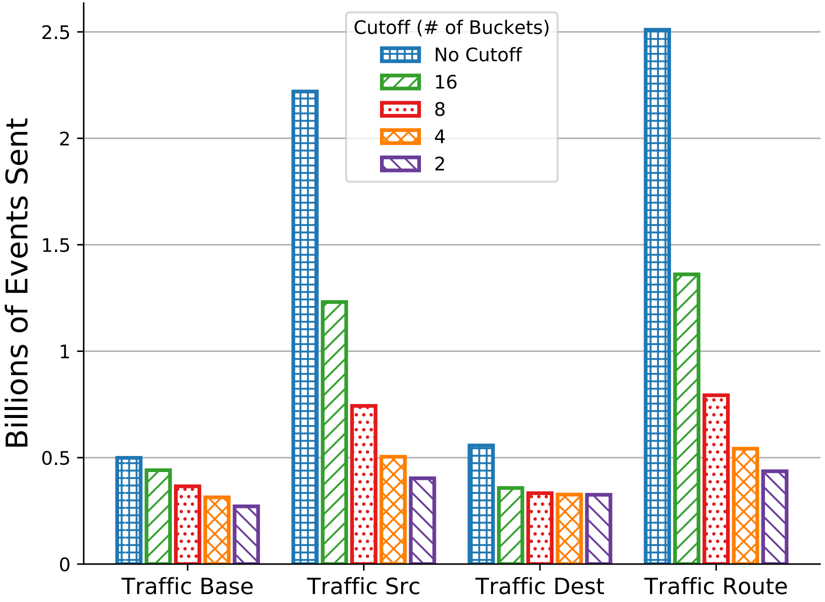


Adaptive Event Throttling

Dragonfly Remote Events

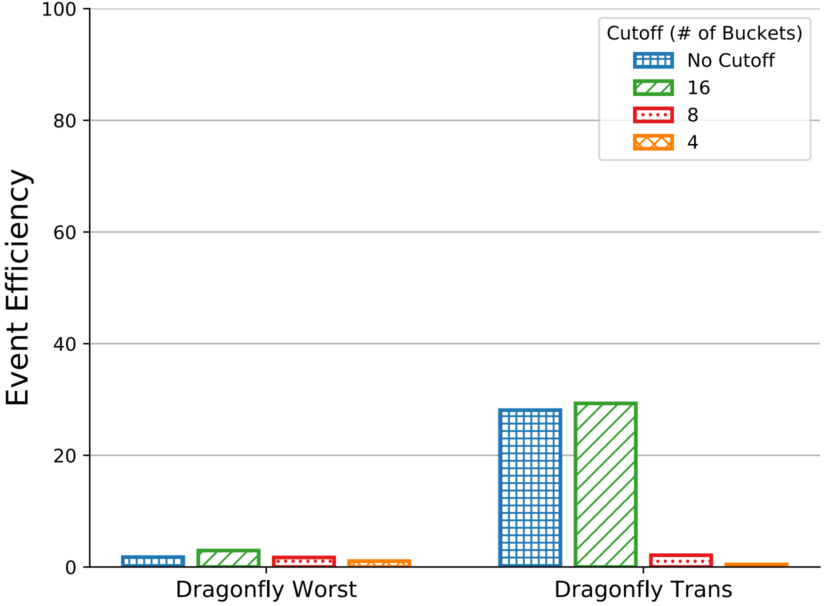


Traffic Remote Events

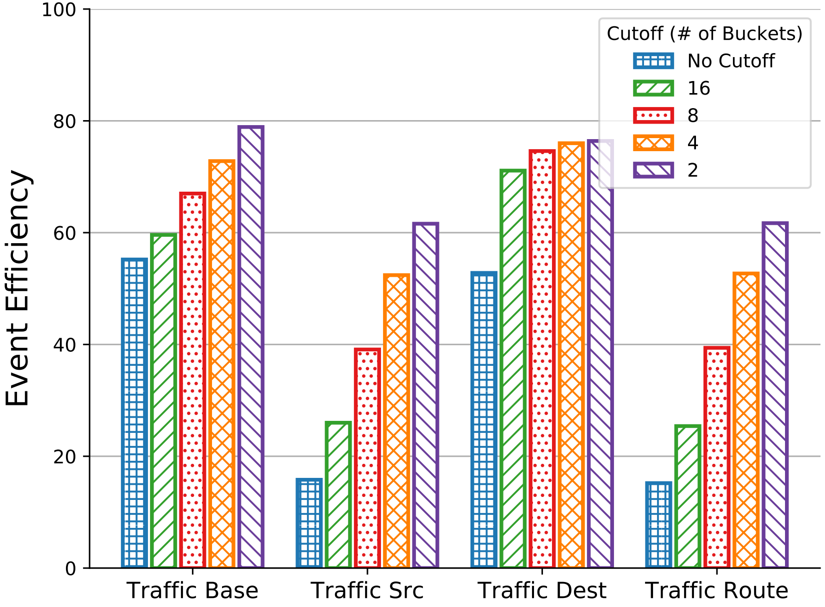


Adaptive Event Throttling

Dragonfly Event Efficiency



Traffic Event Efficiency



How does this differ from SPEEDES?

SPEEDES

- Throttling ***required*** for the GVT computation to complete
- Once throttling starts, ***all events*** are all held until next GVT cycle

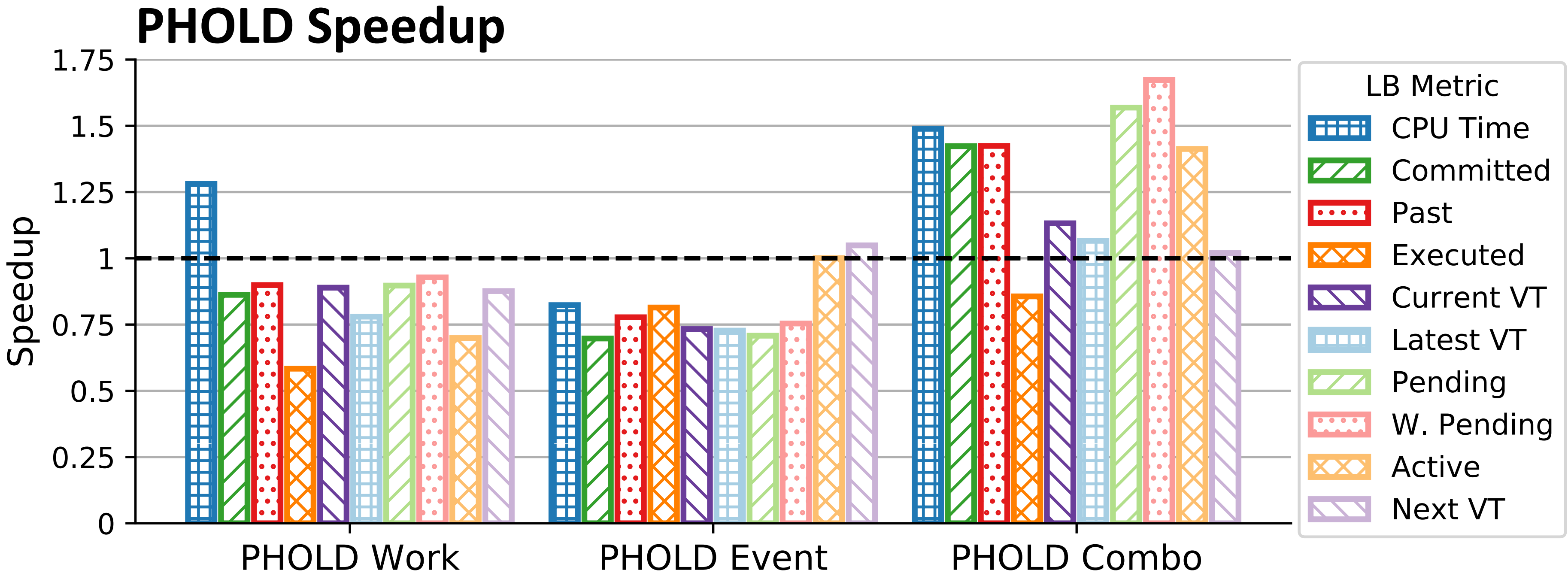
CHARADES

- GVT computation runs regardless of messages in flight – throttling just to improve performance
- Choice to hold an event is per event – holding one does not preclude us from sending another

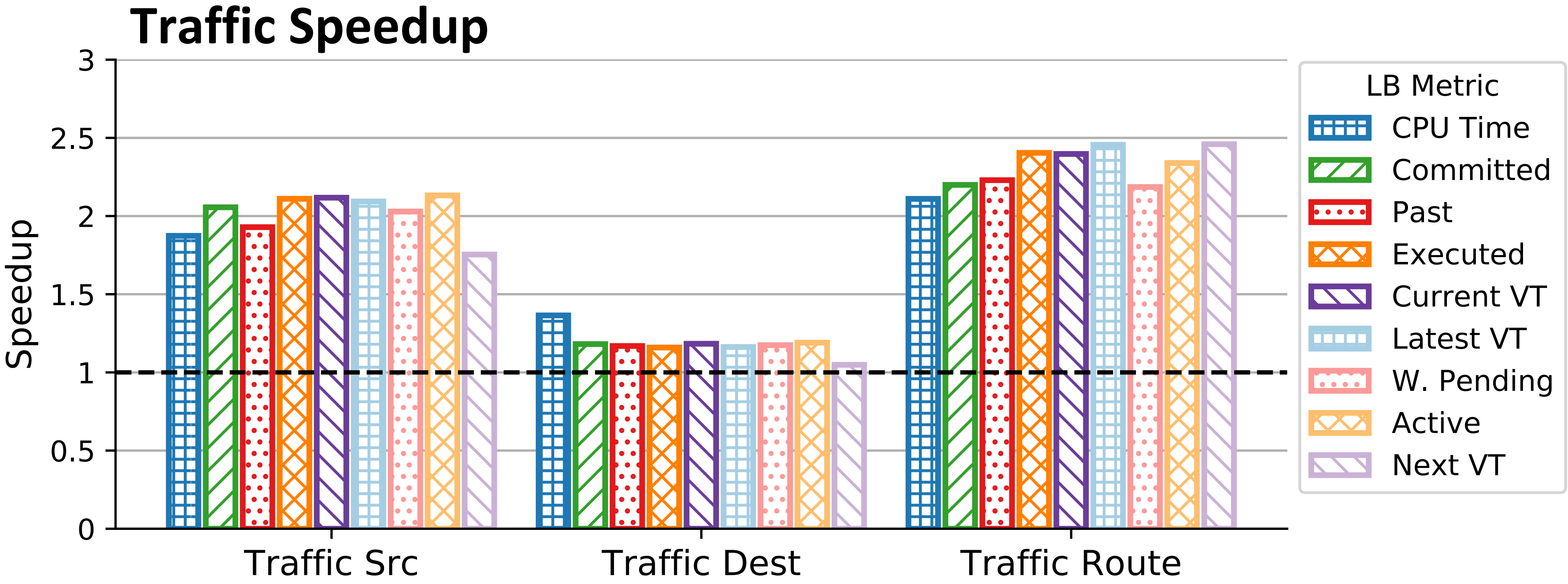
Load Balancing with Bucketed GVT

- Don’t want to stop the simulation
- No obvious synchronization points
  - GVTManager runs independently of Scheduler
- Exploit anytime migration in Charm++
- Throttling improves event efficiency to aid LB

Load Balancing with Bucketed GVT



Load Balancing with Bucketed GVT



Summary

- Proposed the Adaptive Bucketed GVT algorithm
  - Timestamp aware to adapt to simulation conditions
  - Less communication required
  - Allows for adaptive communication throttling
- Load balancing *can* improve event efficiency
  - Metric effectiveness depends on model
- Best performance comes with decoupled solution
  - GVT: sync cost, Throttling: event efficiency, LB: balance

Future Work

- On-line tuning for adaptive event throttling
- Lightweight graph partitioning strategies
- Vectors of load metrics
- ML for load metrics

THANK YOU!